A RANDOM NETWORK IDENTIFICATION MODEL ON DISTRIBUTED AND COMPLEX NETWORKS

U. Mohan Srinivas, *Assoc. Prof., Dept. of CSE & MCA, QIS College of Engineering and Technology, Ongole,*

Sunku. Sai Ranganatha Harish, *Final Year Student of MCA, QIS College of Engineering and Technology, Ongole*

Abstract—*Network discovery assumes a essential job in the investigation of the auxiliary highlights of complex systems. Since on the web systems become progressively expansive and complex after some time, the strategies customarily utilized for network revelation can't proficiently handle expansive scale arrange information. This presents the vital issue of how to successfully and productively find huge networks from complex systems. In this investigation, we propose a quick parallel network disclosure demonstrate called picas (a parallel network disclosure calculation dependent on rough enhancement), which incorporates two new strategies: (1) Mountain model, which works by using diagram hypothesis to surmised the choice of hubs required for combining, and (2) Landslide calculation, which is utilized to refresh the measured quality augmentation dependent on the approximated enhancement. Furthermore, the GraphX circulation registering system is utilized so as to accomplish parallel network discovery over complex systems. In the proposed model, grouping on seclusion is used to instate the Mountain show just as to register the heaviness of each edge in the systems. The connections among the networks are then improved by applying the Landslide calculation, which enables us to acquire the network structures of the complex systems. Broad analyses were led on genuine and manufactured complex system datasets, and the outcomes exhibit that the proposed calculation can beat the cutting-edge strategies, in adequacy and proficiency, when attempting to take care of the issue of network recognition. In addition, we decisively demonstrate that general time execution approximates to multiple times quicker than comparable methodologies. Viably our outcomes recommend another worldview for extensive scale network revelation of complex systems.*

***Index Terms*—Community discovery, complex networks, distributed computing.**

I. INTRODUCTION

Complex systems have turned out to be omnipresent in our day by day life, Such models incorporate online in informal organizations, distribution reference systems, client exchange systems, etc. Because of the perplexing connections among hubs, and the huge cardinality of systems, these systems are alluded to as "intricate system" [1]. Network structure, which starts from complex systems, alludes to a gathering of hubs which are collected into firmly associated gatherings, where there is a high thickness of inside gathering edges and a lower thickness of between-amass edges [2]. It is vital for the motivations behind research to comprehend the auxiliary highlights, the development of networks, the engendering of data, focal points proposal, and other huge highlights. Network disclosure is a standout amongst the most imperative and principal errands in system examination and has applications in practical forecast in Biology [3]. Early research in network revelation for complex systems centers essentially around little systems with basic structures, this is because of the computational troubles of putting away and investigating extensive scale hub and edge data.

Our research is motivated by the following observations: (1) as social networks become more and more embedded in our everyday lives, this intuitively has led to a critical mass of users, e.g., there are 13.5 billions users being active in Facebook each month [4]. With the growth of social networks, traditional community detection algorithms do not scale to the large number of users, the complex relationships between them or the rapid flux their relationships. (2) These increasingly complex and undetected features of large social networks represent missed opportunities for analyzing, correlating, and ultimately predicting the behavior of the users for the purposes of marketing, advertisement and internet public opinion control. (3) The study of the inner and intra structural features of communities in large-scale complex networks has direct practical theoretical applications. And such applications necessitate efficient and accurate algorithms. (4) There exists some parallelized community detection algorithm proposed to process large-scale data.

II. RELATED WORK

With the expanded prominence and pervasiveness of complex systems, the territory of research including the examination of auxiliary highlights inside these systems proceeds to earn more consideration. There have been a few fundamental network discovery calculations proposed since the origin of this territory of research, e.g., Newman et al. proposed the GN calculation [2], the Fast-Newman calculation in view of the possibility of measured quality enhancement [6] and the CNM calculation [7]. These techniques have been generally utilized in recognizing networks in systems [8]. All together to improve the effectiveness of network location, Quiet al. [9] divided the networks utilizing the ghastly cut technique, and the Laplacian grid. Runa et al. [10] introduced a basic methodology of consolidating substance and connection data in diagram structures. Wu et al. [11] proposed a question one-sided hub weighting plan to decrease the unimportant sub-charts and quicken network discovery. All the more as of late, Zhang et al. [12] prescribed upgrades to the CNM calculation by improving the refresh procedure of measured quality. Prat-Perez et al. [13] proposed the weighted network bunching model, which takes the triangle, rather than the edge, as the insignificant auxiliary theme, which shows the nearness of a solid connection in a chart. Ferreira et al. [14] proposed a technique which works to change a lot of time arrangement information into a practically identical arrange utilizing different separation capacities, so as to recognize gatherings of unequivocally associated hubs in complex systems. Shan et al. [15] planned a covering network look system for gathering questions. Huang et al. [16] defined the network location as a issue of finding the nearest bracket network. Li et al. [17] proposed a system to decide networks in a multi-dimensional system dependent on the likelihood dispersion of each measurement figured from the system. To make the procedure of network revelation more powerful, Mahmood et al. [18] proposed an inadequate unearthly bunching calculation dependent on ℓ1 standard limitations to discover a network mark for every hub. Whang et al. [19] proposed a proficient covering network identification calculation utilizing a seed development approach. The previously mentioned strategies for network recognition have demonstrated fundamental in progressing both the zones of research and application, anyway they don't address a basic issue, of which we endeavor to address in this examination, of dealing with huge scale complex system information in a viable and productive way. Dinette al. [20] proposed an added substance guess calculation for measured quality bunching with a consistent factor and they demonstrated that a network structure with particularity subjective near greatest measured quality might bear no likeness to the ideal network structure of most extreme measured quality. Shiokawa et al. [21] proposed a quick particularity based diagram bunching calculation by gradually pruning superfluous vertices/edges and advancing the request of vertex choices. It requires as it were 156 seconds on a chart with 100 million hubs and 1 billion edges. In an unexpected way, picasis a parallel calculation by applying two techniques, i.e., the Mountain Model and the Landslide technique, which can help get high recognition precision with the certification of good runtime execution. So as to address the trouble of preparing system information, which for the reasons for this exploration can be thought about Big Data, parallel calculations were used. Prat-P'erez et al. [22] proposed a high caliber, adaptable and parallel network location approach for substantial diagrams. Be that as it may, because of specific confinements, it isn't suitable for distinguishing covering networks. Wickramarachchi et al. [5] introduced a productive way to deal with identifying networks in huge scale diagrams by improving the consecutive Louvain calculation and parallelizing it on the MPI structure. Viramas et al. [23] proposed an inner circle permeation calculation (CMP) in light of MapReduce to meet the fundamental prerequisites of memory, CPU and I/O activities. The outcomes exhibit that when the number of hubs are more noteworthy than forty thousand, the execution time surpasses 1,000 seconds. As of late, Staudt et al. [24] parallelized the Louvain strategy to effectively find networks in enormous systems. Moon et al. [25] used vertex-driven with MapReduce and Graph Chi to recognize extensive charts in informal communities. Lu et al. [26] proposed a conductance-based network discovery calculation for weighted systems, and structured a productive information sending calculation for defer tolerant systems. Qiao et al. [27] proposed a parallel calculation for identifying networks in complex systems dependent on particularity, and structured new network union and refresh techniques. The parallel diagram grouping models can be connected to identify networks. Meyerhenke et al. [28] proposed a powerful parallel method to segment substantial charts of complex systems. Takahashi et al. [29] proposed a novel calculation SCAN-XP that performs over Intel Xeon Phi to bunch huge scale diagrams. In [30], an intelligent and adaptable diagram bunching calculation on multi-center CPUs was introduced. Evade et al. [31] parallelized huge numbers of diagram grouping calculations in the mutual memory multicore setting. Be that as it may, the proposed diagram bunching models can't be straightly connected to distinguish networks due to complex connections between hubs in complex systems. So as to address these key difficulties, the proficient disclosure of networks, and in a convenient what's more, proficient way, in this exploration we propose a novel network recognition show dependent on estimated streamlining, which is parallelized on the GraphX system [32] to guarantee quick calculation. Whenever analyzed with customary calculations, and parallel calculations, we exhibit that there is a reasonable and quantifiable increment in time execution. Also, forecast precision for this strategy is kept up at an extremely abnormal state.

III. EXISTING SYSTEM

Early research in community discovery for complex networks focuses primarily on small networks with simple structures; this is due to the computational difficulties of storing and analyzing large-scale node and edge information. There exists some parallelized community detection algorithm proposed to process large-scale data. The work done by Wickramarachchi et al. Show that they can achieve fivefold performance improvements when using 128 parallel processors, but in turn requires even more resources to process larger networks

A) Disadvantages

As social networks become more and more embedded in our everyday lives, this intuitively has led to a critical mass of users, e.g., there are 13.5 billions users being active in Facebook each month . With the growth of social networks, traditional community detection algorithms do not scale to the large number of users, the complex relationships. However, due to certain limitations in previous systems, it is not appropriate for detecting overlapping communities.

IV. PROPOSED SYSTEM

We propose Picasso, which is a new community detection model that is much faster than the most state of the art solutions, and improves the quality of community detection. Utilize graph theory for approximate optimization techniques in discovering large communities in complex networks. This is accomplished by taking into full consideration the structural features of communities, and in turn proposing new concepts and algorithms including:

1) the boundary nodes,

2) the chain group for storing the weight of nodes,

3) the Mountain model for choosing nodes to combine, and

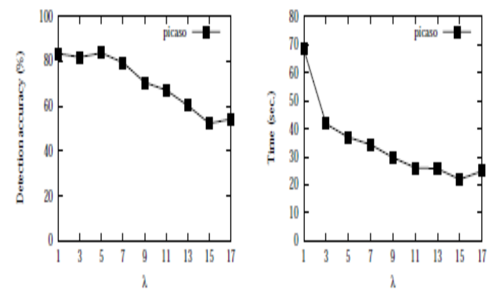
4) the Landslide algorithm used for updating the weights of the chain-group structure and the nodes in communities of the entire network. With the goal of efficiently processing large-scale network data, we propose Picasso that is a parallel community discovery algorithm integrating the Mountain model and Landslide algorithm.

A) *Advantages*

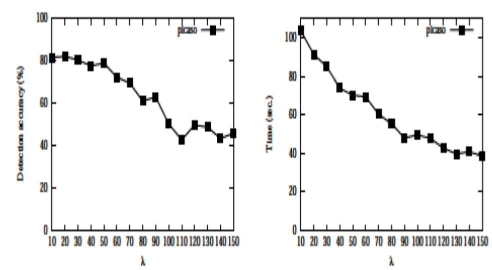
Picaso can handle big complex networks, while traditional serial detection algorithms do not work. We have presented a parallel community discovery algorithm for large-scale complex networks, named picaso. Picaso functions by integrating multiple innovations, which include the Mountain model, a new update strategy called the Landslide algorithm, which is based on approximate optimization techniques and graph theory.

V. PARAMETER STANDARDIZATION

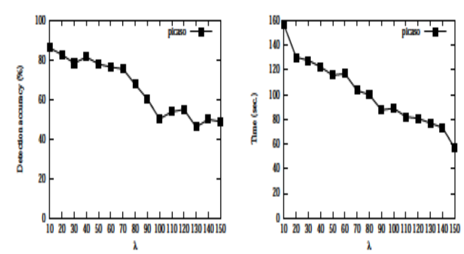
In a trial to form the comparison between the assorted algorithms utilized in the experiments, the parameter λ used in the Mountain model was adjusted consequently. Picaso chooses chain teams at the highest of Mountain model to roughly merge into communities by victimisation the parameter λ, so the choice of λ becomes integral to performance. during this set of experiments, it are observed varied the worth of λ for picaso will have a distinct effects on the district attorney and execution time. The results of this experimentation square measure.



(a) Accuracy on the facebook dataset (b) Efficiency on the facebook dataset



(c) Accuracy on the v-10w datasets (d) Efficiency on the v-10w dataset



(e) Accuracy on com-DBLP dataset (f) Efficiency on com-DBLP dataset

According to Fig. four λ will increase, the prosecuting officer of picaso gradually decreases beneath completely different datasets. In distinction, execution time seems to be reducing within the method. This is as a result of picaso chooses chain teams to merge that have boundary nodes that minimize discrimination among communities. particularly, once the peak of the Mountain model becomes low, picaso might opt for too several chain groups to merge, that may increase the possibility that associate degree incorrect community partition. once λ grows, there are more chain teams to be hand-picked, so the procedure resources are often totally utilised, and also the range of merging operations are often greatly reduced.

VI. THE MOUNTAIN MODEL

A) Basic Concepts:

A complex network is a graph with non-trivial topological features, it has the following properties: self-organization, self-similarity, small world, and scale-free. Fig. 1 is an example of a network with twelve nodes and twenty three edges derived from a complex network.

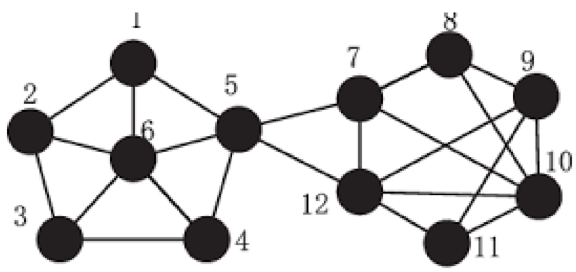


Fig: Example of a sample network

Definition one (Chain Group). a sequence cluster is denoted byCG=, wherever s is that the begin node, t is that the finish node,and r is that the weight between s and t, or the relation kind.It is value to notice that we tend to use the chain-group structureto store the elementary network knowledge in GraphX.Definition a pair of (Boundary Node). Given thatBN=vi ∈ C, vj∈ C′, evivj∈ E representsthe set of boundary nodes, where vi, vj area unit distinctnodes from the communities C and C′, associated evivj is anedge in the sting set E.

$$Q = \frac{1}{2m} \sum_{i,j \in V} (e_{ij} - \frac{d_i d_j}{2m}) \delta(c_i, c_j)$$

Given that relationships between communities isrelatively troublesome to spot from the worldwide perspective, itfollows that Equation one is additionally troublesome to calculate. Newmanproposed a simplified equation as shown below [2].

$$Q = \frac{1}{2m} \sum_{i=1}^{n} (e_{ii} - \frac{d_i^2}{2m})$$

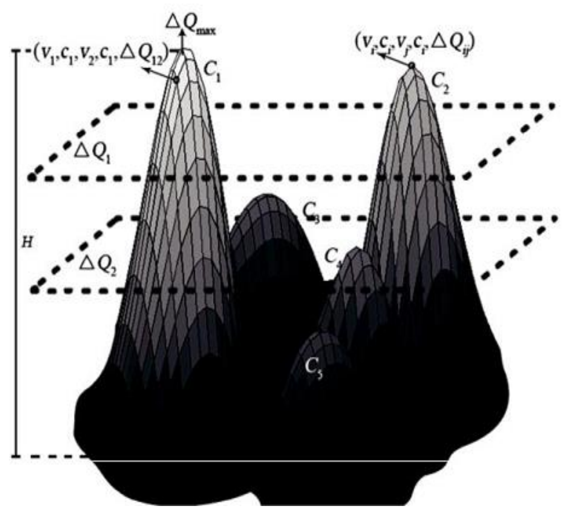
B) The Mountain Model

The Mountain model is integral in this research, and is based on modularity, approximate optimization, and grap theory.

$$\Delta Q = \frac{1}{m} (e_{ij} - \frac{d_i d_j}{2m})$$

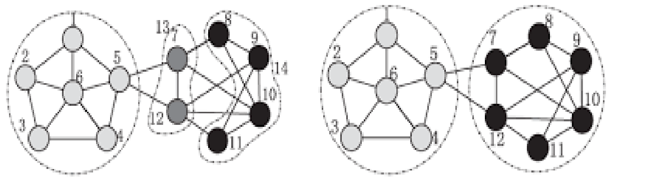
It sorts the chain groups by the weights of edges. Owing to the feature of community structures, some chain groups in a community may fall down while surrounding community may rise like mountains. Resolutely, a suitable number of chain groups at the top of mountains are chosen to form new communities. Resolutely, a suitable number of chain groups at the top of mountains are chosen to form new communities.

Fig: Example of the Mountain model.



C) Parallel Community Detection Model

When the number of nodes and edges in the networks remain unchanged, after the community merging operation, the number of edges in the new community equals the sum of the edges in and between the two merged communities. Moreover, the number of edges between the new community and the other communities equals the sum of edges between the merged communities and other communities..



(a) Before merging the community 13 and 14 (b) After merging the community 13 and 14

The picaso algorithm is designed on the GraphX framework, but it cannot support the attributes of edges and nodes. In order to handle this problem, we store the node set V using a tuple (v, c) in picaso, where v represents the index of a node, and c is the index of the community which v belongsto.

$$a_i = \frac{d_i}{2m}$$

$$a_Z = \sum_{i \in Z} a_i$$

$$a_Y = \sum_{j \in Y} a_j ; Y \not\subset X$$

In addition, picaso stores the edge set E using a triplet (s, t,ΔQ), where s is the start node, t is the end node. The chain group can be obtained by computing the Cartesianproduct of V and E. The essential steps of the picaso algorithm include: (1) parameter initialization, (2) building the Mountain model, (3) merging the nodes and updating, and (4) community generation. Note that, in the first step, the network data is loaded and stored in memory, duplicated edges areeliminated, and the indexes of nodes are reordered.

$$\Delta Q = \frac{1}{2m^2}(2m * \sum_{u \in X, v \in Y} e_{uv} - \sum_{u \in X} d_u * \sum_{v \in Y} d_v)$$

D) Parameter Initialization

In this phase, the task is to calculate the parameters for modularity incrementation w.r.t. chain groups, i.e., the number of nodes n , the degree of each node denoted by

d, the number of edges m, and ΔQ.

Algorithm 1 Parameter initialization

Input: The preprocessed network N.

Output: A graph G.

1. G=graph Loader(D);
2. m=get Edges(G);
3. n=getNodes(G);
4. disseminate m to each machine;
5. for each node i∈ V do
6. di=getDegree(G, i);
7. cId=i;
8. T=V × E;
9. for each t ∈ T do
10. ΔQij = 2 ∗(eij/2m− did/ 4m2);
11. output G;

As shown in Algorithm 1, the first step is to load the network data into memory (line 1), then calculate the number of edges m (line 2) and the number of nodes n (line 3) and disseminate m to each machine (line 4). The second step is to compute the degree of each node (lines 5-6), and specifies the node's community index to be its node index (line 7). The third step is to form chain groups by using the Cartesian product of V and E (line 8), which determines ΔQ w.r.t the chain group (lines 9-10). Lastly, the new graphG is outputted (line 11).

E) Constructing the Mountain model

After initializing the chain group, the Mountain model is constructed, which works to sort the chain groups by their ΔQ. According to Definition 5 and Corollary 1, it is known that the peak of each mountain is mutually-exclusive, thus suitable chain groups are chosen for merging at the top ofthe mountains to form smaller communities with an acceptable λ parameter. The new index is allocated to the newcommunity. The algorithm is given below:

Algorithm 2 Mountain model construction

Input: The graph G=(V,E).

Output: A preliminarily dividing community set C = (C1,C2,C3, ...).

1. H=getHeight(G);
2. λ=2 ∗ |E|/|C|;
3. CG=V × E;
4. for each t ∈ CG do
5. if getAttr(t)≥ ΔQλ then
6. V T=insert(t);
7. for V T/= ∅ do
8. n=n + 1;
9. S′=connectComponent(S);
10. C=insert(n, S′);
11. S=remove(S, S′);
12. output C;

The basic idea of Algorithm 2 is given as follows:

(1) Obtain the maximum height of mountains based on Definition 4 (line 1), compute the parameter λ, and determine the validity of λ (line2).

(2) Obtain the chain group set CG by the taking Cartesian product of V and E (line 3).

(3) Choose the chain groups in CG where ΔQ ≥ ΔQΛ,and form a new set S (lines 4-6).

(4) Compute the connect component of S, where nodes in the same connect component belong to the same community (line 9). Allocate a new index for the newly-formed community (line 10), remove the nodes that have been allocated (line 11), and output the preliminarily dividing community set C (line 12).

VII. RESULT

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.Based on the future security issues, security can be improved using emerging technologies like single sign-on.

VIII. CONCLUSION

In this research, we have presented a parallel community discovery algorithm for large-scale complex networks, named picaso. Picaso functions by integrating multiple innovations, which include the Mountain model, a new update strategy called the Landslide algorithm, which is based on approximate optimization techniques and graph theory. Picaso functions by finding the nodes that meet the condition of aggregation based on the Mountain model, then forms new communities and calculates the modularity increment between the newly formed communities and other communities. Future work will include the Experiments to test the validity of the proposed methods were conducted on synthetic and real large-scale complex network datasets. The resultsdemonstrate that picaso is more effective and efficient on detecting big communities in complex networks and addressing the case when the size of network nodes and edges become extremely large, e.g., more than 1 billion nodes. The proposed algorithm cannot guarantee real time performance in such a case, and will necessitate further innovations to produce efficiency computing of the modularity increment. Another challengethat will be addressed in future work is overlapping community recognition. This will require new methods for which will likely be implemented on the Spark platform. In conclusion, the methods proposed in this research work to contribute to a larger effort targeted at advancing the study of complex community evolution. Understanding the evolution of network structures, analysing, processing and ultimately predicting the behavior of participants in large-scale social networks has and will continue.

IX. REFERENCES

[1] A. Barabasi, R. Albert, H. Jeong, and G. Bianconi, "Power-law distribution of the world wide web," Science, vol. 287, no. 5461,Art. No. 2115, 2000.

[2] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical Review E, vol. 69, no. 2, Art. No. 026113, 2004.

[3] J. Lee, S. P. Gross, and J. Lee, "Improved network community structure improves function prediction," Scientific Reports, vol. 3,no. 2, Art. No. 2197, 2013.

[4] Wearesocial, "Gigital in 2016," 2016. [Online]. Available: http://www.wearesocial.com

[5] C. Wickramaarachchi, M. Frincuy, P. Small, and V. K. Prasannay, "Fast parallel algorithm for unfolding of communities in large graphs," in Proceedings of 2014 IEEE High Performance Extreme Computing Conference. IEEE, 2014, pp. 1–6.

[6] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," Physical Review E, vol. 69, Art. No. 066133,2004.

[7] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," Physical Review E, vol. 70, no. 2, Art. No. 066111, 2004.

[8] M. E. J. Newman, Networks: an introduction. Oxford,UK: Oxford University Press, 2010.

[9] J. Qiu, J. Peng, and Y. Zhai, "Network community detection based on spectral clustering," in Proceedings of the 2014 International Conference on Machine Learning and Cybernetics. IEEE, 2014, pp.648–652.

[10] Y. Ruan, D. Fuhry, and S. Parthasarathy, "Efficient community detection in large networks using content and links," in Proceedings of the 22nd International Conference on WorldWideWeb. ACM, 2013,pp. 1089–1098.

[11] Y. Wu, R. Jin, J. Li, and X. Zhang, "Robust local community detection: on free rider effect and its elimination," Proceedings of VLDB Endowment, vol. 8, no. 7, pp. 798–809, 2015.

[12] X. Zhang, H. You, W. Zhu, S. Qiao, J. Li, Z. Zhang, and X. Fan, "Overlapping community identification approach in online social networks," Physica A, vol. 421, pp. 233–248, 2015.

[13] A. Prat-P´erez, D. Dominguez-Sal, J.-M. Brunat, and J.-L. Larriba-Pey, "Put three and three together: triangle-driven community detection," ACM Transactions on Knowledge Discovery from Data, vol. 10, no. 3, Art. No. 22, 2016.

[14] L. N. Ferreira and L. Zhao, "Time series clustering via community detection in networks," Information Sciences, vol. 326, pp. 227–242, 2016.

[15] J. Shan, D. Shen, T. Nie, Y. Kou, and G. Yu, "Searching overlapping communities for group query," World Wide Web, vol. 19, no. 6, pp. 1179–1202, 2016.

[16] X. Huang, L. V. S. Lakshmanan, J. X. Yu, and H. Cheng, "Approximate closest community search in networks," Proceedings of the VLDB Endowment, vol. 9, no. 4, pp. 276–287, 2015.

[17] X. Li, M. K. Ng, and Y. Ye, "MultiComm: finding community structure in multi-dimensional networks," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 4, pp. 929–941, 2014.

[18] A. Mahmood and M. Small, "Subspace based network community detection using sparse linear coding," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 3, pp. 801–812, 2016.

[19] J. Whang, D. Gleich, and I. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 5, pp.1272–1284, 2016.

[20] T. N. Dinh, X. Li, and M. T. Thai, "Network clustering via maximizing modularity: Approximation algorithms and theoretical limits," in Proceedings of the 2015 IEEE International Conference on Data Mining. IEEE, 2015, pp. 101–110.

[21] H. Shiokawa, Y. Fujiwara, and M. Onizuka, "Fast algorithm for modularity-based graph clustering," in Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence. AAAI, 2013, pp. 1170–1176.

[22] A. Prat-P´erez, D. Dominguez-Sal, and J.-L. Larriba-Pey, "High quality, scalable and parallel community detection for large real graphs," in Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014, pp. 225–236.

[23] A. Varamesh, M. K. Akbari, M. Fereiduni, S. Sharifian, and A. Bagheri, "Distributed clique percolation based community detection on social networks using MapReduce," in Proceedings of2013 5th Conference on Information and Knowledge Technology. IEEE, 2013, pp. 478–483.

[24] C. L. Staudt and H. Meyerhenke, "Engineering parallel algorithms for community detection in massive networks," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 1, pp. 171–184, 2016.

[25] S. Moon, J. G. Lee, M. Kang, M. Choy, and J. W. Lee, "Parallel community detection on large graphs with mapreduce and graphchi," Data and Knowledge Engineering, vol. 104, pp. 17–31,2016.

[26] Z. Lu, X. Sun, Y. Wen, G. Cao, and T. L. Porta, "Algorithms and applications for community detection in weighted networks,"IEEE Transactions on Parallel Distributed Systems, vol. 26, no. 11, pp.2916–2926, 2015.

[27] S. Qiao, J. Guo, N. Han, X. Zhang, C. Yuan, and C. Tang, "Parallel algorithm for discovering communities in large-scale complex networks," Chinese Journal of Computers, vol. 40, no. 3, pp. 687–700,2017.

[28] H. Meyerhenke, P. Sanders, and C. Schulz, "Parallel graphpartitioning for complex networks," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 9, pp. 2625–2638, 2017.

[29] T. Takahashi, H. Shiokawa, and H. Kitagawa, "SCAN-XP:Parallel structural graph clustering algorithm on intel xeon phi coprocessors," in Proceedings of the 2nd International Workshop on Network Data Analytics. ACM, Art. No. 6, 2017.

[30] S. T. Mai, M. S. Dieu, I. Assent, J. Jacobsen, J. Kristensen, andM. Birk, "Scalable and interactive graph clustering algorithm onmulticorecpus," in Proceedings of 33rd IEEE International Conference on Data Engineering. IEEE, 2017, pp. 349–360.

[31] J. Shun, F. Roosta-Khorasani, K. Fountoulakis, and M. W.Mahoney, "Parallel local graph clustering," Proceedings of the VLDBEndowment, vol. 9, no. 12, pp. 1041–1052, 2016.

[32] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin,and I. Stoica, "GraphX: Graph processing in a distributed dataflowframework." in Proceedings of the 11th USENIX Symposium onOperating Systems Design and Implementation, vol. 14. USENIXAssociation, 2014, pp. 599–613.

[33] S. F. A. Lancichinetti, "Limits of modularity maximization incommunity detection," Physical Review E, vol. 84, no. 6, Art. No.066122, 2011.

[34] J. Leskovec, "SNAP: Stanford large network dataset collection,"2016.[Online].Available:http://snap.stanford. edu/data/index.html

[35] M. E. J. Newman, "The structure and function of complexnetworks," SIAM Review, vol. 45, no. 2, pp. 247–256, 2003.